



|  |    |
|--|----|
| Forenings_oplysninger .....                              | 2  |
| Generalforsamling .....                                  | 3  |
| Bestyrelsesmøde / Resumé .....                           | 5  |
| Enquete / resultat .....                                 | 6  |
| Opgave-2 .....   | 7  |
| 2 x FORTH .....  | 9  |
| AutoCAD tip 'Out of RAM' .....                           | 12 |
| PC-SIG 575 PC-stock .....                                | 13 |
| - 576 Pc tickle .....                                    | -  |
| - 577 C tutor (disk 1 of 2) .....                        | 14 |
| - 578 C tutor (- 2 - 2) .....                            | -  |
| - 579 Pascal tutor (disk 1 of 2) .....                   | 15 |
| - 580 Pascal tutor (- 2 - 2) .....                       | -  |
| - 581 Wordworker (disk 1 of 2) .....                     | -  |
| - 582 Wordworker (- 2 - 2) .....                         | 16 |
| - 583 Lotus 123 - the whiterock alternative .....        | -  |
| - 584 Miscellaneous lotus 123 worksheets .....           | -  |
| - 585 Dos extensions (disk 1 of 2) .....                 | 17 |
| - 586 Dos extensions (- 2 - 2) .....                     | -  |
| - 587 Symphony word processing tips & macros .....       | 18 |
| - 588 Symphony bank/insurance/medical applications ..... | -  |
| - 589 Ptools .....                                       | 19 |
| - 590 Nuclear Magnetic Resonance (nmr) .....             | 20 |
| - 591 Genesis ver 1.0 .....                              | -  |
| - 592 Tshell .....                                       | -  |
| Turbo Pascal 5.5 .....                                   | 21 |
| 1 x FORTH .....  | 25 |
| Adresser mv. ....  | 31 |

# MØDER

Der er planlagt møder følgende datoer:

onsdag 20 sep 1989

Modula-2 Shareware Compiler demo og gennemgang af et simpelt Modula-2 program. Compiler med utilities kan hentes fra BBS systemet, kan rekvireres fra formanden eller fra biblioteket. 16 Color Low resolution graphics mode for CGA.

onsdag 4 okt 1989

Generalforsamling, (Se andetsteds.) Grafik & spil. Diverse flysimulatorer vil for denne ene gangs skyld blive vist på et medlemsmøde.

onsdag 8 nov 1989

Object oriented programming, C++ og Turbo Pascal 5.5.

onsdag 6 dec 1989

Henrik Persson: Maskin orienteret C-programmering, memory residente programmer på PC.

Møderne afholdes på adressen:

Vesterbro Kulturhus, Lyrskovsgade 4, København V.

Lokale-nummer fremgår af opslag ved indgangen. Dørene åbnes som vanligt kl. 19.

## INDMELDELSE OG KONTINGENT

Indmeldelse i MUG Denmark kan foretages ved indbetaling af kontingent (pt. 200 kr. årligt) på Giro 5 68 65 12, MUG Denmark, ved Lars Gråbæk. Der opkræves ikke indmeldelsesgebyr.

Oplag: 300  
Tryk: Dansk Tidsskrifts Tryk.  
Redaktion: Viggo Jørgensen.

Redaktionen afsluttet 5/9-89.

Fra bestyrelsen  
INDKALDELSE TIL GENERALFORSAMLING

Der indkaldes til ordinær generalforsamling i MUG

onsdag den 4 okt. 1989

kl. 19.00

i

Vesterbro Kulturhus, Lyrskovsgade 4, København V

Dagsorden:

1. Valg af dirigent og referent.
2. Formandens beretning.
3. Bibliotekarernes beretning.
4. Indkomne forslag:  
    Annoncering af møder i ungdoms-milieu'er med henblik på  
    at aktivere nogle unge mennesker i bestyrelsesarbejdet.
5. Valg af formand.
6. Valg af bestyrelsesmedlemmer.  
    CSL genopstiller ikke.  
    Der er m.a.o. enkelte ledige pladser i bestyrelsen ...
7. Valg af 2 revisorer.
8. Eventuelt.
9. Indlæg om grafik programmering med turbo pascal 5,  
    low resolution - men 16 farver,  
    samme teknik som spillet MOON-bug m.fl.

Der vil desuden blive WORKSHOP med demonstration af PC spil.

En kommentar fra formanden:

Ad 2.

Om den forløbne periode kan man konkludere, at der har været en række positive aktiviteter, på trods af problemer med besættelse af bestyrelsesposter, uenighed i bestyrelsen om betaling for særligt krævende bestyrelsesarbejde (som fx bladudgivelse).

Bladet er vågnet op, er så at sige blevet renoveret af Viggo Jørgensen, som dog mener, at der er alt for få leverandører til bladet.

På generalforsamlingen vil jeg imidlertid gerne have afsat flere midler til bladet.

C-programmerings kurset havde enorm tilstrømning, kurset var interessant for både nye og gamle, frafaldet var ikke urimeligt stort.

BBS systemet kører jo både for medlemmer og ikke medlemmer, og må anses for at være et fast holdepunkt i tilværelsen for en hel del computer-freaks, eller computer brugere, som er lidt foran.

Interessant, at man kan have et system, som henter, sorterer og evt. udskriver post til morgen (eller aften-) kaffen. Og interessant, at det ikke er statsdrevet, ikke får offentligt tilskud etc. (Det er der forresten tanker om at lave om på.)

Ad 9.

Spil er for så vidt udmærkede demonstrationer af, hvad man kan med en computer, og for den tænksomme også af hvad man ikke kan.

Imidlertid er jeg principielt meget uinteresset i spil på computere, hellere så jeg en udbredt interesse for og aktivitet omkring PC'en som avanceret værktøj til decentral administration og græsrodsbevægelse.

---

#### Forkortet referat af bestyrelsesmøde onsdag 30.8.89

---

Dagsordenen handlede især om resultat på medlemsenquete, om mødeemner og om mulighed for at få gode foredragsholdere, annoncere møder bredere - men lokalt - og om foreningens karakter, landsdækkende eller lokal, amatør eller fagligt oplysende.

Enqueten, som kun kan tages som en strømpil, viste helt klart størst interesse for avanceret C programmering og objekt orienteret programmering. Det ønske vil vi forsøge at honorere.

Møderne afholdes i Medborgerhuset i Lyrskovgade 4, København V.

Mødeemnerne blev flg.:

20.9 Modula-2 Shareware Compiler demo og gennemgang af et simpelt Modula-2 progr.

Compiler med utilities kan hentes fra BBS systemet,

rekvireres fra formanden eller fra biblioteket.

16 Color Low resolution graphics mode for CGA.

4.10 Generalforsamling, Grafik & spil. Se andetsteds.

Diverse fysimulatorer vil for denne ene gangs skyld blive vist på et medlemsmøde.

8.11 Object Oriented Programming, C++ og Turbo Pascal 5.5.

6.12 Henrik Persson: Maskin orienteret C-programmering, memory residente programmer på PC.

Der blev derefter drøftet regnskab, PC'en til regnskab/medlemskartotek, som på sidste generalforsamling blev skrabet sammen af løse stumper blandt medlemmerne, kan nu langt om længe få de manglende dele.

Bladet prioriteres højt.

Brugertallet på BBS systemet er faldet vanvittigt efter indførelse af nye regler. Det var også meningen.

BBS nettet kan ændres til en tilskudsberettiget forening.

Biblioteket skal fornyes, økonomien er i orden, men vi mente ikke, at man skal købe alle volumes fra PC-SIG i USA, hellere vente på, at folk bestiller, og så forsøge at skaffe de pågældende volumes. Kopiering fra medlemmers samlinger kunne også være en mulighed for fornyelse.

Medlems software er helt gået i stå - undtagen vores pladearkiv.

Foreningens oprindelige MPS-hardware konstruktører må generelt set formodes at være begyndt at finde løsninger i PC-software i stedet for i udskiftning af bus drivere på MPS systemer o.l.

Der blev desuden udvekslet synspunkter om foreningens størrelse, sammenlægning med andre foreninger o.l.

Undertegnede foreslog, at vi opprioriterer landsdækkende aktiviteter, blad og BBS, samt afholdt et landsdækkende møde, gerne sammen med andre foreninger, en "hacker conference" af en slags; at vi annoncerer interessante møder offentligt for at tiltrække unge mennesker, som jo senere kan overtage foreningen med dennes aktiver og muligheder.

En kommentar til foreningens karakter:

Meningen skulle være at danne et fagligt forum for unge såvel som gamle programmører, der ønsker at udveksle meninger og erfaringer.

Dertil kræves et fagligt højt niveau, og vi kan måske forbedre os på dette punkt. Vi kan ikke forventes at være eksperter på alle punkter, det er klart, men kan vi mon ikke gøre det mere attraktivt for eksperter udefra at komme til os og holde foredrag, skrive artikler m.m.?

En forening af den art som ovenfor beskrevet skal have et større medlemstal, det er klart, men jeg mener iøvrigt at interessen er der, hvis blot vi kan strukturere foreningen fornuftigt.



and for now... I'm off to the South Pacific. I need a break.

Statistisk fordeling af indkomne Giro-kort svar  
m.h.t. ønskede forenings-aktiviteter

-----  
Præferencer blandt forslag til møde\_emner:  
-----

|    |      |  |
|----|------|--|
| A  | iiii | Object Oriented Programming                                  |
| A1 | iii  | C++  |
| A2 | i    | Prolog   |
| A3 |      | Small-talk   |
| B  | i    | Administration af større software projekter i Turbo Pascal 5 |
| C  | iiii | Avanceret C - programmering                                  |
| D  | i    | Databaseteknikker for PC'ere                                 |
| E  | ii   | Modula-2   |
| F  | i    | Forth (el. lign.) programmering                              |
| G  |      | Grafik   |
| H  |      | 68000 assembler  |
| I  | i    | intel 8086 asm.prog.   |

-----  
Foretrukne benyttede edb værktøjer  
-----

|    |        |                     |
|----|--------|---------------------|
| J  | iiii   | Loddekolbe          |
| K  | iii    | Assembler           |
| K1 | ii     | 8086                |
| K2 |        | 68000               |
| K3 | i      | Z80                 |
| L  | iiii   | Regneark            |
| M  | iii    | Tekstbehandling     |
| M1 | i      | m.grafik            |
| N  | iiiiii | Modula-2, C, Pascal |
| O  |        | Cobol               |
| P  |        | Prolog              |

-----  
andet: Dataflex, Turbobasic  
-----

-----  
MEST benyttede maskintype  
-----

|    |        |      |                        |
|----|--------|------|------------------------|
| Q  | iiiiii | iiii | PC'er                  |
| Q2 | iiiiii |      | 286ere                 |
| Q5 | i      |      | Nec'er                 |
| R  | iii    |      | Z80/8080               |
| S  | ii     |      | CP/M 2.2               |
| T  |        |      | Atari/Amiga            |
| U  |        |      | Apple                  |
| V  |        |      | IBM type mainframe     |
| W  | i      |      | DEC                    |
| X  | i      |      | Andre mainframe typer. |

## OPGAVER

Til opgaven i sidste nummer (om addition af et tal og dets omvendte, til resultatet bliver et tal, som læses ens forfra og bagfra) indkom kun én løsning, som til gengæld var rigtig: tallet 89 og dets omvendte 98 skal gennemgå processen 24 gange, før man får det ønskede resultat. Ingen andre talpar kræver mere end 6 additioner.

Der blev altså ikke brug for lodtrækning mellem indsenderne. Vinderen er Kurt Mielke, som har løst opgaven ved hjælp af et program i C.

Præmien, et volume efter eget valg fra et af diskettebibliotekerne, kan fås ved henvendelse til bibliotekaren.

Kurts program og løsning er vist

### Den ny opgave:

På et skakbræt er felterne nummereret med bogstaverne A-H på den ene led og med tallene 1-8 på den anden.

En springer i skak bevæger sig i en lidt speciel gangart, idet den i hvert træk flyttes 2 skridt i én retning og 1 skridt vinkelret derpå.

Fra feltet F5 kan en springer således flyttes til et af felterne E7, G7, H6, H4, G3, E3, D4 eller D6.

*På et skakbræt med kun 4x4 felter, nummereret A-D og 1-4, placeres en springer på feltet A1. Den skal ved træk som de ovenfor beskrevne flyttes til A4. Hvad er det største antal felter, den kan passere undervejs (A1 og A4 medregnet), når hvert felt kun må passeres én gang?*

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 8 |   |   |   |   |   |   |   |   |
| 7 |   |   |   |   |   |   |   |   |
| 6 |   |   |   |   |   |   |   |   |
| 5 |   |   |   |   | S |   |   |   |
| 4 |   |   |   |   |   |   |   |   |
| 3 |   |   |   |   |   |   |   |   |
| 2 |   |   |   |   |   |   |   |   |
| 1 |   |   |   |   |   |   |   |   |
|   | A | B | C | D | E | F | G | H |

|   |   |   |   |   |
|---|---|---|---|---|
| 4 |   |   |   |   |
| 3 |   |   |   |   |
| 2 |   |   |   |   |
| 1 |   |   |   |   |
|   | A | B | C | D |

Løsninger sendes til

Niels Veileborg  
Finsensvej 11 B 2.th.  
2000 Frederiksberg

senest den 24 oktober 1989.

Vinderen findes om nødvendigt ved lodtrækning. Præmien er som sidst et volume efter eget valg fra brugergruppens diskettebiblioteker.

```

/*-----*/
/ talleg.c May 14 15:03 1989. Kurt Mielke, besv. af konkurrence: MUG 1989:2 /
/*-----*/

extern double atof();
double reverse();

main() {
    int tmp, max = 0;
    double i, maxi;

    for ( i = 10; i<100 ; i += 1.0 ) {          /* check between 10 and 99 */
        tmp = check(i,0,0);                    /* save no. of additions */
        if (tmp > max) {                       /* if bigger than previous */
            max = tmp;                          /* save new */
            maxi = i;
        }
    }
    printf("%0.0f skal skal \"address\" %d gange:\n",maxi,max) ;
    check(maxi, 0, 1);
}

check(i,count,verb) /* recursive function, returns number of additions */
double i;
int count,verb;    /* if verb tells what we are doing */
{
    double j,sum;

    j = reverse(i);          /* find the reverse */
    sum = i+j;
    if (verb)
        printf("%0.0f + %0.0f = %0.0f\n",i,j,sum);
    if (sum == reverse(sum) ) /* is sum equal to it's own reverse? */
        return count + 1;    /* then we are through */
    check(sum,count+1,verb); /* else do another addition */
}

double reverse(x)
double x;
{
    char s[50], r[51];
    char *rp, *sp;

    sprintf(s,"%0.0f",x);    /* convert x to a string */
    sp = s;                  /* sp points to beginning of s */
    rp = r + sizeof(r);     /* rp - - end of r */
    *--rp = '\0';           /* zeroterminate r; */
    while (*sp)              /* while chars in s */
        *--rp = *sp++;      /* copy in reverse order */
    return atof(rp);        /* convert to integer and return it */
}

```

Run

89

(skal "address" 24 gange)

89 + 98 = 187

187 + 781 = 968

.

.

1801200002107 + 7012000021081 = 8813200023188



## MVP-Forth fra MUG BBS

og

## UNI Forth fra Databankens BBS.

*Af Donald Axel*

Hvorfor Forth? Artiklens forfatter afstår ganske fra at besvare dette i øvrigt udmærkede spørgsmål; artiklen handler om to Public Domain Forth systemer, som hver især er interessante.

MVPFORTH (Mountain View Press Forth) kan hentes fra MUGBBS eller rekvireres fra MUG biblioteket (mod betaling); UNIFORTH fandt undertegnede hos Bjørn Løndahl, Databanken.

MVPFORTH kunne jeg ikke downloade ("hente") i sin fulde længde, jeg formoder jeg er stødt ind i en softwarefejl, eftersom dusinvis af andre downloads har fungeret uden problemer. Efter andet forsøg på en alternativ maskine opgav jeg, og må derfor nøjes med at kommentere den del, som kom helskindet over. Det er dog ikke den mindst interessante, det er nemlig kilde tekst i assembler til en FORTH kerne.

Med lidt tålmodighed kan den læses, rettes til efter behov og formentlig forbedres hist og her. Den kan ikke køre på normal DOS uden videre, men må kunne bringes til det, eftersom problemerne er ret begrænsede: kommandoen Bye (exit til OS) crasher, formentlig p.g.a. forkerte segment value i DS. Der er nok også en fejl eller i hvert fald uheldsmæssigheder i filhandlingen (efter DOS 1.00 typen).

Kildeteksten består af ca. 57 Kb ASM; MASM version 1.00 kan klare at tygge igennem uden rettelser, og som derefter kan linkes af de fleste linkere, da den jo ikke kræver noget library. MVPFORTH.EXE bliver på lidt over 10Kb.

MVPFORTH inkluderer iflg. kildetekst kommentarerne i begyndelsen alle F79 ord (The entire Forth-79 required word set) Forfatteren har også skrevet en bog, "ALL ABOUT FORTH", som skulle kunne købes fra MVP:

Mountain View Press, PO Box 4656, Mountain View, CA 94040, (U.S.A.)

Et Forth subprogram (i Forth verdenen kaldet et "ord") bestående af maskin instruktioner skal kobles til den resterende del af Forth elementerne ved at installere sig i et område af Forth procedure area og efterlade et "visitkort" i form af navn plus adresse i et dictionary.

Næsten alle Forth kommandoer (eller "ord") holder til i dictionary. Dette begynder i lav memory, 3-400 bytes efter starten af programmet. Dictionary består af 3 sektioner.

Kernen: Assembly language program(mer) som indeholder 190 primitive funktioner.

Grundlæggende Forth: System definitioner eller subprogrammer bygget over de første "kernefunktioner".

Applikations funktioner: Definitioner (subprogrammer) og/eller "programmet" som brugeren har skrevet (eller har fået skrevet) Området vokser opad, efterhånden som bruger/programmør kommer med nye definitioner.

Illustration 1 som iøvrigt er hentet fra UniForth manualen, viser en FORTH memorymap. Der er ikke, så vidt jeg har kunnet konstatere, forskel på UniForth memory layout og MVPFORTH. Begge udspringer af en z80 Forth, som vistnok er resultatet af kreativiteten i en "Forth Interest Group", hvoraf navnet FIG Forth.

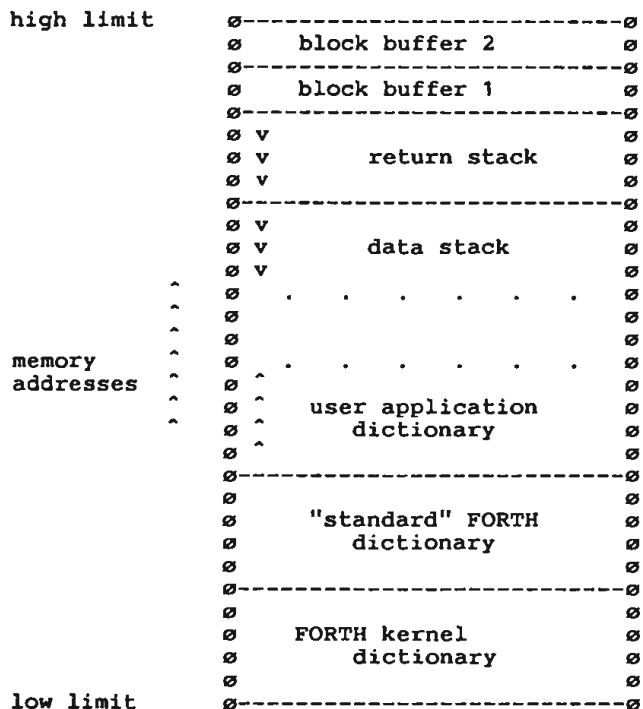


ILLUSTRATION 1

De tre dele af dictionary ses i low memory. Der er egentlig ingen funktionel grund til at skelne mellem "kerne" og "standard" og "user application" - dictionary. Forskellen ligger i abstraktionsniveau og maskin-interface: "kernen" er maskinkoblingen, "standard" er koblingen til andre "Forth" brugere "user application" området høster resultatet: brugeren kan i bedste fald programmere sin application på samme måde som på en anden maskine med en andre peripherals og - ja - en anden CPU.

User application området kan sagtens benyttes til fx. at definere "Forth" ord fra andre "Forth" standarder eller andre maskinkoblinger o.s.v. - funktions-principperne er de samme som for kerne og standard.

Det er jo interessant i mange sammenhænge. Problemet er naturligvis, at "standard" er lidt flydende. Ud fra de to modtagne pakker dog ikke værre, end at man kunne leve med det.

Lad mig sige med det samme, at for mig er FORTH ikke interessant på længere sigt, medmindre nogen kan overbevise mig om at overhead (maskintidsforbrug) er mindre end på optimerede assembly language programmer, SAMT at man kunne komme til at skrive sine Forth programmer på en måde, der er virkelig læselig.

Men tanken er selvfølgelig fascinerende: Mainframes fra 7 forskellige leverandører med transaktionssystemer mm. programmerede med FORTH programmer, som for applikationsprogrammøren er nøjagtig ens på de 7 maskiner.

I "kerne-" såvel som brugerdefinerede assembly language funktioner skal følgende registerkonventioner overholdes:

```
;FORTH REGISTERS
;
;FORTH 8088 FORTH PRESERVATION RULES
;
; IP SI INTERPRETER POINTER.
; MUST BE PRESERVED
; ACROSS FORTH WORDS.
;
; W DX WORKING REGISTER.
; JUMP TO 'DPUSH' WILL
; PUSH CONTENTS ONTO THE
; PARAMETER STACK BEFORE
; EXECUTING 'APUSH'.
;
; SP SP PARAMETER STACK POINTER.
; MUST BE PRESERVED ACROSS
; FORTH WORDS. THIS IS ALSO
; THE SYSTEM STACK, WHICH
; IS SOMETIMES CALLED THE
; HARDWARE STACK -HS .
;
; RP BP RETURN STACK POINTER.
; MUST BE PRESERVED ACROSS
; FORTH WORDS.
;
;IN ADDITION, THE 8088 CODE SEGMENT, DATA
;SEGMENT AND STACK SEGMENT REGISTERS ( CS,
;DS, AND SS, MUST BE PRESERVED ACROSS
;FORTH WORDS.
;
;THE OTHER REGISTERS OF THE 8088 CPU MAY
;BE USED FREELY.
;
;10 AUGUST 1982
```

ILLUSTRATION 2

Det er nu på tide at se hvad data-stakken er for noget: Det er brugerens "LIFO" last in first out område, hvor fx. mellemregningstal gemmes. Dette kendes fra kalkulatorer som benytter "omvendt polsk notation":

```
ønsker man at beregne 2 + 2 skrives man
2 <enter>
2 <enter>
plus <enter> ( display viser 4 )
```

Normale kalkulatorer benytter jo en anden indtastningsteknik, men benytter rent teknisk samme "lifo" system:

```
2 <plus>
2 <=> ( display viser 4 )
```

Udskrift fra en Forth beregning af  $2*2 + 310$ :

```
2 2 * 310 + . <enter> 314 ok
```

Stakken er som andre stakke ofte er, vendt på hovedet, vokser altså fra høj memory ned mod lavere memory (se illustration 1).

Retur stakken, som anvendes meget forsigtigt, indeholder adresser på subfunktioner. Anvendes desuden for loop control variables, og for variable, som iøvrigt er "i vejen" på datastakken. Returstakken er typisk meget lille, 160 bytes i UniForth. (I MVP Forth kan man jo selv udvide denne).

Mountain View Press Forth må være mere interessant end her beskrevet, hvis man kan få dokumentationen med. Ønsker man at se assembler source på et højt niveau til en Forth interpreter kerne, så er MVPFORTH sagen. MVPFORTH har ikke floating point arithmetic.

## UNIFORTH

Det har til gengæld UniForth. Den har stort set det hele, men kildeteksten "leveres" ikke til UniForth; har man behov for det, kan man anskaffe den "professionelle" pakke, som leveres med 400 kb kildetekst.

Kildeteksten til "kernel" sælges IKKE. Den er på 5 kb. Derfor har Mountain View Press pakken alligevel interesse for den, som vil se på Forth med en PC-kirurgs røntgenblik. En nattehacker og en ordentlig disassembler vil naturligvis kunne give visse informationer om denne "kernel", men det er nok vanskeligt at få det store overblik over 1000 - 1500 liniers systemprogram uden kommenteret source.

Forskellen på Public Domain "Sampler" pakken og den "professionelle" pakke:

Multitasking, interproces kommunikation, 50% hurtigere, string package, kbd access, tone (music) generation, hard/soft float, CGA grafik: punkt, streg, fyldte polygoner; hurtig video editor, source til alt undtagen kernen, debugger, decompiler, flytbarhed til flere systemer.

Iflg. dokumentationen er prisen for UNIFORTH "prof" 245\$ - men der er gået et stykke tid siden denne pris, som er fra aug. 1985. En 80286 vers. annonceredes for 295\$.

Et eksempel (fra manualen) på anvendelse er et system med en user task, en temperatur kontrol, printer spooler og telekommunikation kørende samtidigt og uden nævneværdig degradation af performance.

UniForth indeholder desuden nogle produkt-specifikke Forth extensions: Kontrol flow muligheder med de gængse benævnelser "while", "repeat" m.m. men - naturligvis - med et lidt andet indhold.

...næste gang: **Forth set fra tastaturet:**

---

## **AutoCAD / "Out of RAM"**

Brug af Autocad til 2-D tegneopgaver giver som regel ingen problemer, heller ikke med 3-D tegninger af kompleksitet som fx "Peters Kirkens" skelet (gitterkonstruktion), men vil man fjerne "hidden lines" kan man få en melding som Out of RAM... og har man ikke lige det pågældende manual-afsnit liden for arms rækkevidde kan gode råd være dyre.

AutoCAD har/bruger et lager-område til 'calculations' mv. I mange versioner har dette en default størrelse på omkr. 14 K, ofte er det ikke tilstrækkeligt og nævnte melding fremkommer, hvorefter programmet tager en pause (lang pause!) - filen lukkes.

En hurtig afhjælpning er at gå til DOS og skrive: set acadfreeram = 15 det øger 'work space' til 15 K, måske skal der mere til, prøv at øge med 1 K indtil ok. det rltige må findes ved forsøg.

Denne set operation skal gøres efter hver DOS start! Kommandoen kan naturligvis også anbringes i en AUTOEXEC.BAT.

-----  
DISK NO 575 PC-stock  
-----

PC-stock finance program is a stock evaluation and tracking system.

PC-STOCK EXE Stock trend analysis progr. (128k + colorgraphicard)  
PC-STOCK DOC Documentation  
PC-STOCK INF Over lay program for pc-stock  
README Documentation (brief) for printing manual  
DAYTON STK Stock trend files  
DJ20BND STK - - -  
DJIND STK - - -  
INTHARV STK - - -  
MESABI STK - - -  
MIDSUT STK - - -  
PANAM STK - - -  
S&P20T STK - - -  
S&P400I STK - - -  
S&P40FIN STK - - -  
S&P40UT STK - - -  
S&P500 STK - - -  
TESORO STK - - -  
AUTOEXEC BAT Batch file to automatically start PC-STOCK.EXE  
SILVER STK Silver trend files  
GOLD STK Gold trend files

-----  
DISK NO 576 PC tickle  
-----

PC-tickle is an appointments calendar/checkbook manager/record keeper program that can help the user organize his schedule by tracking important appointments & dates, with reminders at the appropriate time. It also keeps track of checkbook expenditures, and can even be used as a calorie counter.

CAL BAT Batch file to print calendar file  
CALENDAR Textfile of calendar appointments  
CALORIES - with calorie consumption records  
CHECKS - checking account records  
EXPENSES - expense records  
HOLIDAYS - dates of holidays  
PCBOOK COM Progr. to read documentation on screen interactively  
PCBOOK HLP Help screens for reading documentation on screen interactively  
PCBOOK MNU Menus to select portions of documentation to read  
PCLTR0 EXE Auxiliary program to tickled  
PRINTCAL BAT Batch file to print calendar  
PRINTDOC BAT Batch file to print documentation  
READD0C BAT Batch file to display documentation on screen  
README Introduction - how to print documentation or read it on screen  
README BAT Batch file to display readme file on screen  
TICKLE COM Main progr. - appointment calendar/checkbook manager  
.TICKLE DOC Encrypted documentation file for pc tickle  
TICKLED COM Mini editor program to edit tickle files

-----  
DISK NO 577 C tutor (disk 1 of 2)  
-----

This is the 1st disk of the C tutor system. It is a comprehensive introductory tutorial to programming in the c language. This disk contains the textual tutorial material in a series of .TXT files.

----- C tutor

|           |     |   |
|-----------|-----|---|
| READ      | ME  | Short program description & printing instructions |
| PRINTTEXT | BAT | Batch file to print all .TXT files                |
| TABCONT   | TXT | Tutorial table of contents                        |
| INTRO     | TXT | Tutorial introduction                             |
| CHAP1     | TXT | Tutorial chapter 1                                |
| CHAP2     | TXT | " " 2   |
| CHAP3     | TXT | " " 3   |
| CHAP4     | TXT | " " 4   |
| CHAP5     | TXT | " " 5   |
| CHAP6     | TXT | " " 6   |
| CHAP7     | TXT | " " 7   |
| CHAP8     | TXT | " " 8   |
| CHAP9     | TXT | " " 9   |
| CHAP10    | TXT | " " 10  |
| CHAP11    | TXT | " " 11  |
| CHAP12    | TXT | " " 12  |
| CHAP13    | TXT | " " 13  |
| CHAP14    | TXT | " " 14  |
| LIST      | COM | Program to print files                            |

-----  
DISK NO 578 C tutor (disk 2 of 2)  
-----

It is the 2nd disk of the C tutor system. It contains the example C programs as discussed in the text portion of the tutorial.

----- C tutor

|          |     |  |
|----------|-----|--|
| READ     | ME  | Short text file to describe progr.& printing process |
| COMPILER | DOC | Discussion of various compilers                      |
| PRINTALL | BAT | Batch file to print out all source c files           |
| ???????? | C   | 76 source code files for miscellaneous applications  |
| test     | bat |  |
| LIST     | EXE | Program to list source files                         |
| STRUCT   | DEF | Data file structure definition                       |
| VC       | EXE | Executable calculator file                           |
| VC       | DOC | Short documentation about the calculator program     |
| HELP     |     | Calculations text file                               |
| AMORT    | -   | - -  |
| PAYMENT  | -   | - -  |
| TEST     | -   | - -  |

-----  
DISK NO 579 Pascal tutor (disk 1 of 2)  
-----

Pascal tutor is a paper based introductory tutorial on the use of the pascal language. The tutorial is distributed on diskette & can be printed conveniently. The second disk contains the pascal source code for examples used throughout the tutorial.

Disk #1

----- Pascal tutor  
READ ME Text file describing product & printing procedures  
PRINTTEXT BAT Batch file to print all of the .TXT files  
TABCONT TXT Tutorial table of contents  
INTRO TXT Tutorial introduction  
CHAP1 TXT Tutorial chapter 1  
CHAP2 TXT " " 2  
CHAP3 TXT " " 3  
CHAP4 TXT " " 4  
CHAP5 TXT " " 5  
CHAP6 TXT " " 6  
CHAP7 TXT " " 7  
CHAP8 TXT " " 8  
CHAP9 TXT " " 9  
CHAP10 TXT " " 10  
CHAP11 TXT " " 11  
CHAP12 TXT " " 12  
CHAP13 TXT " " 13

-----  
DISK NO 580 Pascal tutor (disk 2 of 2)  
-----

This is the second disk of the pascal tutor system. It contains the pascal source code for all of the examples discussed in the text based tutorial as contained on disk #1.

----- Pascal tutor  
READ ME Short description of programs and printing process  
PRINTALL BAT Batch file to print source code files  
???????? PAS 67 source files  
REALDATA TXT Data file  
OT DOC Notes on the ot program  
OT PAS Source code for the ot program  
OT COM Executable ot program  
LIST COM Dos program to list files

-----  
DISK NO 581 Wordworker (disk 1 of 2)  
-----

Wordworker is a cross-reference progr. For the new testament with over 14,800 cross references to 7956 vers. It can also provide the number of occurrences for over 6063 words. This disk also contains a complete editor to generate letters, sermons or other text with the information contained in the set.

|          |     |   |
|----------|-----|---|
| ALPHSHEL |     | Alpha-shell file                            |
| EXT      | DCT | List of words - 2                           |
| INT      | DCT | List of words - 1                           |
| ORDERFRM |     | Order form                                  |
| PROFILE  | WWP | Batch file                                  |
| PUNFILE  |     | Punctuation file                            |
| README   |     | Introduction to system files & installation |
| REFFILE  |     | External cross reference                    |
| RESPONSE |     | Response form                               |
| WW       | COM | Main wordworker program                     |
| WW       | DEF | File with different term definitions        |
| WW       | HLP | Explanation of function keys & operations   |
| WWSETUP  | COM | Wordworker setup program                    |
| XPOINT   |     | Cross point program for system              |
| XREF     |     | Cross reference program                     |

-----  
DISK NO 582 Wordworker (disk 2 of 2)  
-----

Wordworker is a cross-reference progr. For the new testament with over 14,800 cross references to 7956 vers. It can also provide the number of occurrences for over 6063 words. This disk also contains the data disk for disk number 581.

-----  
DISK NO 583 Lotus 123 - the Whiterock alternative  
-----

A user-friendly file link front end to lotus to allow easier access and use of worksheet files. Several applications working under this environment are provided.

|          |     |   |
|----------|-----|---|
| AUTO123  | WKS | Auto-run file - whiterock altern. menu & file driver. |
| LIBRARY  | WKS | Library of lotus 123 macros                           |
| CHEKBOOK | WKS | Checkbook ledger worksheet                            |
| LOANTABL | WKS | Installment loan amortizer worksheet                  |
| MAILLIST | WKS | Mailing list maintenance worksheet                    |
| WEEKRPT  | WKS | Jobcosting time management worksheet                  |

-----  
DISK NO 584 Miscellaneous Lotus 123 worksheets  
-----

An immense variety of Lotus 123 worksheets can be found on this disk, ranging from printing utilities to financial applications, scientifically oriented spreadsheets to perform polynomial regressions & triangle problems. There are also worksheets for time management, mailing lists and many more.

|          |     |   |
|----------|-----|---|
| ASCIUISE | WKS | How to use chars2 worksheet.                          |
| CHARS2   | WKS | File of ascii chars not directly accessible by Lotus_ |
| CHEKBOOK | WKS | Personal checkbook ledger. 123.                       |
| CODESAMP | WKS | Printer driver for specialized fonts.                 |
| CONTENTS | WKS | Toolkit shell and ascii driver.                       |



|          |     |   |
|----------|-----|---|
| LOAN     | WKS | Loan analysis worksheet.                              |
| LOANTABL | WKS | Installment loan analyzer.                            |
| MAILLIST | WKS | Mailing list database.                                |
| NEWCOLOR | IN  | Redirect output for debug progr. to change Lotus 123_ |
| NEWCOLOR | WKS | Instructions for NEWCOLOR.IN. _colors.                |
| POLREG   | WKS | Calculate polynomial regressions.                     |
| PRTGRAPH | WKS | Instructions for using PRTGRAPH.COM for spec. fonts.  |
| TITLES   | WKS | Prepare chronological dates as titles for columns.    |
| TRIANGLE | WKS | Solve for triangles given sides.                      |
| VENDOC   | WKS | Documentation for VENDORS.WKS                         |
| VENDORS  | WKS | Macro driven data base of suppliers for business.     |
| WEEKRPT  | WKS | Time management/job logging utility.                  |

-----  
DISK NO 585 Dos extensions (disk 1 of 2)  
-----

This disk contains the user manual, reference manual and one of a series of extremely flexible & useful dos extensions. These do not necessarily replace resident dos commands, which can still be used as is. Rather, they add more versatile and powerful supersets of these commands which greatly add to their utility.

|      |     |   |
|------|-----|---|
| LS   | EXE | Dos extension - directory lister                    |
| REF  | DOC | Documentation - reference manual for dos extensions |
| USER | DOC | - - manual for install. & use of extensions         |

-----  
DISK NO 586 Dos extensions (disk 2 of 2)  
-----

This disk is the second of the series. It contains all but one of the actual dos extensions in this package.

|        |     |   |
|--------|-----|---|
| ATT    | EXE | Dos extension - view or change file attributes      |
| ALARM  | EXE | Dos ext. - sound a controllable tone                |
| CAT    | EXE | Dos - - concatenate files                           |
| CLK    | EXE | Dos - - measure elapse time of progr.s or commands  |
| CMD    | EXE | Dos - - execute progr.s, commands, or batch files   |
| CP     | EXE | Dos - - copy files                                  |
| DIRSRT | EXE | Dos - - directory sorter                            |
| GPM    | EXE | Dos - - find text patterns in files                 |
| INFO   | EXE | Dos - - display system, disk, or memory information |
| INP    | EXE | Dos - - input data to a dos pipe                    |
| MERGE  | EXE | Dos - - merge text lines from multiple files        |
| MV     | EXE | Dos - - move files                                  |
| PRNT   | EXE | Dos - - output selected text lines of files         |
| RENDIR | EXE | Dos - - rename a subdirectory                       |
| RENM   | EXE | Dos - - rename files                                |
| RM     | EXE | Dos - - delete files                                |
| SED    | EXE | Dos - - file editor                                 |
| SELECT | EXE | Dos - - visual selector                             |
| SLEEP  | EXE | Dos - - pause command                               |
| STAT   | EXE | Dos - - statistical text file information           |
| TEE    | EXE | Dos - - dos pipe output controller                  |
| TXLAT  | EXE | Dos - - character translator                        |

|         |     |          |                                 |
|---------|-----|----------|---------------------------------|
| UNDEL   | EXE | Dos ext. | - retrieve a deleted file       |
| UNIQ    | EXE | Dos      | - - duplicate text line handler |
| VOLM    | EXE | Dos      | - - view or change volume name  |
| WHEREIS | EXE | Dos      | - - file locator                |

-----  
DISK NO 587            Symphony word processing tips & macros  
-----

This disk contains a variety of prize-winning tips & macros for symphony, incl. tips on importing wordprocessor files, multiplan files, using sidekick with symphony, and much more.

|          |     |  |
|----------|-----|--|
| TWOCOL   | WRK | Sets up a multi-column document.                               |
| WORDST   | CTF | Converts a wordstar document for symphony.                     |
| WRDPRF   | WRK | Use word proof in conjunction with symphony.                   |
| SEARCH   | APP | Searches spreadsheet for specific character.                   |
| HEX2DE   | WRK | Converts hexadecimal to decimal numbers & vice versa.          |
| CALC2    | WRK | Calculates elapsed calendar & business days between two dates. |
| WNDWSY   | WRK | Simulates 123 window synchronization capability.               |
| RESICO   | WRK | Returns metric values of resistance & tolerance of resistors.  |
| MPLAN    | APP | Imports multiplan data into symphony.                          |
| SEARCH   | WRK | Worksheet using search capability.                             |
| SEARCH   | PRN | Printable documentation on search capability.                  |
| SEARDOC  | WRK | Worksheet format search documentation.                         |
| SKPROB   | WRK | How to use sidekick with symphony.                             |
| ADDRMAC  | WRK | Create an address book.  |
| APPTMENT | WRK | An appointment and time management system.                     |
| DUALDS   | WRK | Install symphony with two monitors                             |
| DEMO     | WRK | Demonstration file.  |
| FLIPPER  | APP | Data mover extension.  |
| LABELS   | WRK | Produces labels based upon a mailing list.                     |
| LEEVFORM | WRK | Offers an important macro concept using .(get).                |
| MACRO1   | WRK | Demonstration macro.   |
| MOVIE    | WRK | Use symphony to display moving graphics.                       |
| ORD_ENT  | WRK | Uses almost all symphony commands for order-entry program.     |
| USERTIPS | DOC | Protocols for use of some of these capabilities.               |

-----  
DISK NO 588            Symphony bank/insurance/medical applications  
-----

The files on this disk demonstrate the uses that Symphony can be put into the banking, insurance, and medical industries.

|          |     |  |
|----------|-----|--|
| 2_APPLIC | WRK | Example file for applic worksheet      |
| 2_ASSET  | WRK | - - - asset worksheet                  |
| 2_BOND   | WRK | - - - bond worksheet                   |
| 2_CASH   | WRK | - - - cash worksheet                   |
| 2_EXCHAN | WRK | - - - exchange worksheet               |
| 2_HUMANR | WRK | - - - humanres worksheet               |
| 2_MORTGA | WRK | - - - mortgage worksheet               |
| ASSET    | WRK | Assists in asset-liability management. |
| AUTODEMO | WRK | Autorun demo                           |

BOND WRK Calculates a bond market price.  
 CASH WRK Forecasts cash flow.  
 DEMO PIC Graphics file for demo.  
 EXCHANGE WRK Effects of exchange rates on loan payments.  
 HUMANRES WRK Organizes employee information.  
 MORTGAGE WRK Determines if customer qualifies for mortgage loan  
 1\_APPLIC WRK Example file for applic worksheet  
 1\_AUDIT WRK - - - audit worksheet  
 1\_CLIENT WRK - - - client worksheet  
 1\_FORECA WRK - - - forecast worksheet  
 1\_IVENTO WRK - - - inventory worksheet  
 1\_LIFE WRK - - - life worksheet  
 AUDIT WRK Calculate appropriate amount of home owner insurance.  
 CLIENT WRK Manage client info. Make follow-up sales call schedule  
 FORECAST WRK Calc. historic.growth figures for insurance products.  
 IVENTORY WRK Maintain, stock and order inventory items.  
 LIFE WRK Estimate apppr. life insurance coverage for client.  
 1\_BUDGET WRK Example file for budget worksheet  
 1\_CAPEQP WRK - - - capeqpt worksheet  
 1\_RESEAR WRK - - - research worksheet  
 1\_SCHIDL WRK - - - schedule worksheet  
 BUDGET WRK Forecast the financ.prospects of a medical practice.  
 CAPEQPT WRK Determine investment quality of potential purchases.  
 DEMI PIC Graphics file for demo  
 RESEARCH WRK Track & graph patient data. Generate reports from  
 findings.  
 SCHEDULE WRK Schedule patient appointments.

-----  
 DISK NO 589 Ptools  
 -----

This Ptools disk comes with many handy routines which may be needed by the serious turbo pascal programmer.

PTOOL1 BOX Combination of ptooldat, ptoolent, ptoolscr programs  
 PTOOLDAT INC Routines to convert & manipulate gregorian & julian\_  
 dates  
 PTOOLDAT PAS Demo of ptooldat routines  
 PTOOLENT DOC Documentation of the ptoolent routines  
 PTOOLENT INC Routines to display,edit & validate data entry fields  
 PTOOLENT PAS Demo of the ptoolent routines  
 PTOOLSCR INC Routines to manipulate screen functions  
 PTOOLSCR PAS Demo for the ptoolscr routines  
 PTOOLTIM PAS Demo for the ptooltim routines  
 PTOOLTIM INC Routines to read and interpret system clock  
 PTOOLWIN DOC Documentation of the ptoolwin routines  
 PTOOLWIN PAS Demo for the ptoolwin routines  
 PTOOLWIN INC Routines to create and manipulate text windows  
 PTOOLWI3 DOC Documentation of the ptoolwi3 routines  
 PTOOLWI3 INC Routines to create and manipulate text windows  
 PTOOLWI3 PAS Demo for the ptoolwi3 routines

-----  
DISK NO 590 Nuclear Magnetic Resonance (NMR)  
-----

Diskette contains a highly technical & useful nmr analysis progr. Nuclear magnetic resonance is used in many advanced chemistry applications and this progr. allows the analysis of this technique. Written by a professor of chemistry at the University of South Florida, this package is unique in the fact that it displays the calculated spectrum including adjustable peak broadening. Simulated plotting of very complex molecules, such as protein spectra are also allowed. Parameters for up to 7 spins (nucleii) can be calculated.

----- NMR submissions  
NMR BAT Batch file to start program  
NMR DOC Short documentation about the program  
NMR? BAS Basic code for program modules 0 - 7  
NMR? EXE Compiled version of module 0 - 7  
SCRATCH MNR Misc data file for program

-----  
DISK NO 591 Genesis ver 1.0  
-----

Genesis is a program designed to work with traveller ( a science fiction game ) from game designers workshop. It helps to speed up the process of setting up planets by the referee & handles aspects such as planetary size, atmosphere & population.

README NEW File with list of revisions  
SETUP BAT Batch file for setup  
SECTOR CHN Planet sctor detail file  
SUBSEC CHN Planet subsector detail file  
PLANET CHN Planet detail file  
GENESIS COM Main genesis program  
SYSTEM1 DAT System data file to set up planet  
GENESIS DOC System documentation with users guide  
AUTOEXEC BAT Auto-execute file  
README NOW Introduction to genesis with feature list  
PLANET 000 Sample planet detail file  
--GNSS10 DID Genesis system work file

-----  
DISK NO 592 Tshell  
-----

Tshell is a powerful visual shell for the pc which provides an efficient environment for the user to manage dos. Version 1.1 is a working demonstration of a larger, more powerful version (2.0).

TSH HLP This file contains help screens  
TSH DOC Documentation manual  
SETUP TSH The customizing routine for Tshell  
README 1ST Contains comments on install. and operation of tshell  
INSTALL BAT Batch file to install tshell on a designated disk  
TSH EXE The main tshell program file.  
USQ EXE Expands (unsqueezes) compressed Tshell files

# Turbo Pascal Version 5.5

Det første systematiske tiltag til at lette løsningen af store programmeringsopgaver blev taget med ALGOL 60, der var det første blokstrukturerede sprog. Pascal er en af dets efterkommere, og underinddelingen af et program i blokke (kaldet procedurer og funktioner i Pascal-sammenhæng) er en afgørende del af løsningen af enhver programmeringsopgave.

I Pascal er der imidlertid ikke faciliteter til at sammenknytte data og algoritmer, og dataenes repræsentation og behandling hænger nøje sammen. Der er heller ikke faciliteter til at angive hvilke operationer, der må udføres på hvilke data. Det giver f.eks. ingen mening at gange to beløb med hinanden, og til at arbejde med punkter i planen i såvel retvinklede som polære koordinater kræves et dobbelt sæt procedurer og funktioner.

Faciliteterne i Turbo Pascal Version 5.5 til arbejde med såkaldte objekter fjerner disse ulemper. Ved et objekt forstås en samling data og algoritmer til at bearbejde dataene, der er knyttet sammen. Algoritmerne lægges på vanlig vis ud i procedurer og funktioner og kaldes i objektsammenhæng metoder.

Programmering med objekter har været kendt i mange år - særlig i LISP - og er også muligt med konventionelle programmeringssprog (J.S.Linowes har vist det for Cs vedkommende i Byte/august 1988), men det har for de konventionelle programmeringssprogs vedkommende været for besværligt til at være af praktisk betydning.

## Dokumentation og hjælp

Dokumentationen består af "Brugerhåndbogen" og "OOP-håndbogen", der er for-danskede og byder på en indføring i Pascal og programmering med objekter, og af "Reference Guide", der er et systematisk opslagsværk på engelsk. "Brugerhåndbogen" og "Reference Guide" er overtaget fra version 5.0.

Oversættelsen af "Brugerhåndbogen" er dårlig, men indholdet er i orden, og det er kun enkelte steder, man falder over fejl, der tyder på, at den sidste korrektur er sprunget over.

Med "OOP-håndbogen" er sagen en anden. Den dækker forholdene vedrørende programmering med objekter og enkelte andre faciliteter, der adskiller version 5.5 fra 5.0. Oversættelsen til dansk er foretaget af IDDoc a.s., Bergen, og Syntax Documents, København.

Mage til makværk af en oversættelse er ikke før set. Der er i snit 10 grammatiske fejl per side, 3 stavfejl og det amerikanske idiom er bibeholdt, så bogen er næsten ulæselig. Således er den skrevet i 2. person indikativ, der, som bl.a. professor J.Lund har gjort opmærksom på, intet har at gøre i faglitteratur på dansk.

Nogle eksempler fra OOP-håndbogen:

"Self" er en automatisk defineret identifikator og hvis du skulle finde dig selv i en identifikator-konflikt inden i en metode,..." (side 18),

"En fremgangsmåde til at give en handling eet navn, der deles op og ned i hierakiet, hvor hver objekt i hierakiet implementerer handlingen på en måde som passer til sig selv." (side 29),

"Destruktørens kode fortjener et nærmere kig." (side 56).

Sproget i Turbo Pascals indbyggede hjælpefunktion er ligeså ringe.

## Hastighedsgevinst

Der blev foretaget en sammenligning af køretiden for programmer oversat med hhv. version 3.0 og 5.5, og de blev 35% hurtigere med version 5.5. Sammenligningsgrundlaget var programmer, der hverken brugte flydende tal eller harddisk særlig meget.

## Fejlmeddelelser

Syntaksanalysen foregår i Turbo Pascal rod-til-blad, hvad der fører til enkelte mærkværdigheder i forbindelse med fejlmeddelelser. Det har også betydning for syntaksen for konstanter, der fra version 5.0 har kunnet angives som beregningsudtryk, og som ikke må begynde med '(' i almindelighed er fejlmeddelelserne udmærkede med en nøjagtig angivelse af fejllens placering i programteksten.

## Brug af objekter

Som det ses i eksemplet fremtræder objekter næsten som poster. Procedurer og funktioner, der i et objekt knyttes sammen med dataene, de skal behandle, kaldes metoder. I eksemplet drejer det sig om 'Constructor Initialiser' og 'Function Laes\_Radius' i typeerklæringerne 'Retvinklet\_type' og 'Polaer\_type'. 'Object(Retvinklet\_type)' i definitionen af 'Polaer\_type' betyder, at 'Polaer\_type' arver alle variabelerklæringer fra 'Retvinklet\_type'. I eksemplet drejer det sig om 'a' og 'b'.

Der skal udføres nogle initialiseringer i den interne administration af objekter, før de kan bruges. Det sker ved at kalde en metode, hvor nøgleordet 'procedure' er erstattet af 'constructor'. Procedurer mærket 'constructor' adskiller sig isøvrigt ikke fra andre procedurer og bruges her til at initialisere abcisse og ordinat hhv. radius og vinkel i objekterne 'Ret\_punkt' og 'Pol\_punkt'.

Bemærk punktumnotationen, der for objekters vedkommende er udvidet til at omfatte funktioner og procedurer - nødvendigt for at skelne metoderne i de to objekttyper fra hinanden.

Funktionen 'Areal' demonstrerer en af fordelene ved programmering med objekter: Koden i 'Areal' er ikke afhængig af repræsentationen af dataene. Metoden 'Laes\_radius', der følger med parameteren 'Punkt' til 'Areal', er knyttet sammen med den aktuelle parameters type, så den rette metode til beregning af cirkelens radius kaldes.

Ved kørsel af programmet bliver resultatet ved begge kald af 'Areal'  $2\pi$ .

## Teknikken bag objekter

Det følgende går på det, der i "OOP-håndbogen" kaldes virtuelle metoder, og som er den type metoder, der har størst interesse.

Reglerne for typekompatibilitet er ændrede, hvad objekter angår. Et objekt kan tilordnes værdien af et objekt af samme type og enhver af dets underordnede typer. Følgelig kan procedurer og funktioner have aktuelle parametre af forskellige objekttyper, og det kan så på oversættelsestidspunktet ikke afgøres hvilke metoder, der skal kaldes via en formel parameter.

For hver objekttype er der en tabel over adresserne på koden for de tilhørende metoder. Det kaldes en VMT-tabel (Virtual Methods Table). Hver objektforekomst indeholder et felt, VMT-feltet, med adressen på den tilsvarende VMT-tabel.

Ved kald af en metode findes først VMT-tabellen via det givne objekts VMT-felt, og i VMT-tabellen kan adressen på koden for den ønskede metode så slås op.

Af afsnittet om objekter som parametre til procedurer og funktioner i "OOP-håndbogen" (side 33) fremgår det, at aktuelle parametre konverteres til de formelle parameters type.

Ved overførsel af værdiparametre medtages VMT-feltet ikke, og metoderne, der bruges, er dem, der hører til den formelle parameters type.

Ved brug af variabelparametre er det imidlertid VMT-feltet fra den aktuelle parameter, der bruges.

Typekonvertering af parametre ved kald af procedurer, hvad der i praksis vil sige, at kun feltene i den formelle parameters type medtages i aktiveringsposten, er bl.a. af effektivitetshensyn påkrævet, men det er en faldgrube, at der m.h.t. metoder er en semantisk forskel mellem variabel- og værdiparametre.

```

Program Ret_Pol;
type Retvinklet_type = object
  a, b: Real;
  Constructor Initialiser(Abcisse, Ordinat: Real);
  Function Laes_Radius: Real; Virtual;
end;

Polaer_type = object(Retvinklet_type)
  Constructor Initialiser(Radius, Vinkel: Real);
  Function Laes_Radius: Real; Virtual;
end;

var Ret_Punkt: Retvinklet_type;
    Pol_Punkt: Polaer_type;

Constructor Retvinklet_type.Initialiser(Abcisse, Ordinat: Real);
begin a := Abcisse; b := Ordinat; end;

Function Retvinklet_type.Laes_Radius: Real;
begin Laes_Radius := sqrt(sqr(a) + sqr(b)); end;

Constructor Polaer_type.Initialiser(Radius, Vinkel: Real);
begin a := Radius; b := Vinkel; end;

Function Polaer_type.Laes_Radius: Real;
begin Laes_Radius := a; end;

(* Parameteren "Punkt" betragtes som et punkt på
   periferien af en cirkel, der har centrum i origo. *)

Function Areal(var Punkt: Retvinklet_type): Real;
begin Areal := pi * sqr(Punkt.Laes_Radius); end;

begin (* Program RetPol *)
  (* Cirkel med radius  $\sqrt{2}$  og centrum
     i origo i kartesiske koordinater *)
  Ret_Punkt.Initialiser(1, 1);
  (* Ditto i polære koordinater *)
  Pol_Punkt.Initialiser(sqrt(2), pi/4);

  writeln(Areal(Ret_punkt):0:4);
  writeln(Areal(Pol_Punkt):0:4);

end.

```



Ovenstående program eksempel ville ikke fungere efter hensigten, hvis parameteren 'Punkt' til funktionen 'Areal' havde været en værdiparameter.

### Konklusion

Turbo Pascal Version 5.5 er et program på højeste faglige niveau, der på mange punkter - oversætterhastighed, fleksibilitet, brugervenlighed og faciliteter til fejlretning m.m. - er en udelt fornøjelse at bruge. Vel er der ikke registeroptimering, men programmerne er hurtige endda.

Som et kuriosum kan det nævnes, at der til nybegyndere er en 'Tour de Pascal', der, som navnet siger, byder på en rundtur til en række af faciliteterne - en fin ide. Enkelte fejl i version 5.0, som bl.a. "Personal Computer World" har gjort opmærksom på, er fjernet i version 5.5.

Hvad dokumentationen og den indbyggede hjælpefunktion angår, må man betakke sig for at få en oversættelse, når det gøres på det niveau, Syntax Documents og program-mørerne hos Borland Scandinavia lægger for dagen. Det frarådes at købe den danske udgave, men løvrigt er Turbo Pascal 5.5 rigeligt pengene værd.

#### Turbo Pascal Version 5.5

Computer: IBM/XT/AT el. 100% kompatibel.  
EGA/VGA kan udnyttes til at få 43/50 linier på skærmen.  
Kapacitet: Min. 448 Kb RAM. DOS 2.0 eller senere.  
Pris u.moms: Standardudgaven: Kr. 1695,-  
Den professionelle udgave: - 2700,-  
Lavere pris ved ombytning med tidl. udgaver af Turbo Pascal  
Importør: Borland Scandinavia, telf.: 42 27 64 55

## Testresultat

### Turbo Pascal Version 5.5

|                       | Dårligt                  | Rimeligt                            | Godt                     | Meget godt                          | Excellent                           |
|-----------------------|--------------------------|-------------------------------------|--------------------------|-------------------------------------|-------------------------------------|
| Fleksibilitet         | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| Dokumentation         | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            |
| Brugervenlighed       | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| Fejlhåndtering        | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| Programmet som helhed | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |

Grep 2. del ( eksempler ), bringes i et senere nr.



## FORTH - for fremtiden

Af Erik Skovgaard, Carlsberg Forskningscenter, Bryggeriteknologisk Afdeling.

De fleste højniveausprog, som benyttes i mini- og mikrocomputere er i virkeligheden oprindeligt udviklet til at køre på store anlæg. De er derfor ofte meget ressourcekrævende, relativt langsomme og klodsede med hensyn til I/O, når de bliver flyttet over i et mikroprocessormiljø.

Rather og Moore indså dette i 1976, da de skulle udvikle et større programmel til et automatisk radioteleskop og skrev derfor programmeringssproget FORTH (1).

De rapporterede i denne forbindelse en opnået programmeringstid på under 10% af den forud estimerede, d.v.s. baseret på traditionelle minicomputer implementerede sprog. Samtidig reduceredes også forbruget af hukommelse fra 64K til 16K. Disse resultater er senere blevet bekræftet via andre opgaver.

I det lys er det besynderligt, at FORTH ikke er blevet mere udbredt end tilfældet er, men forklaringen ligger måske i det noget utraditionelle udseende af programmerne skrevet i dette sprog. Først og fremmest sker alle overførsler af data via en stack og i omvendt polsk notation.

Selvom det i Hewlett-Packard-lomme-regnerens tidsalder ikke burde være et ubekendt begreb, viger de fleste programmører bort fra denne metode til fordel for den funktionelt noget langsommere, men traditionelt mere læsbare matematiske notation.

Dertil kommer, at FORTH ikke opererer med linienumre eller linietags, men derimod med et hierarki af procedurer, der kaldes ved hjælp af navne, som passende kan have noget at gøre med procedurernes funktion: DREJ for positionsbestemmelse og ANTENNE1 for output proceduren for denne bestemte enhed, eller SAML DATA HVER 5 MINUTTER, hvilket forklarer sig selv.

Netop dette, at FORTH er et rent procedureorienteret programmeringssprog, gør, at man opnår en optimal udnyttelse af sine computerressourcer, som vil blive søgt belyst i det følgende.

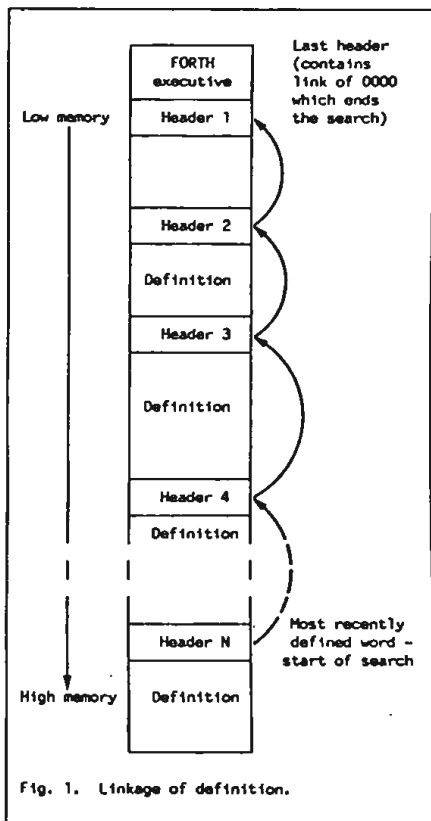


Fig. 1. Linkage of definition.

## Opbygning

FORTH er bygget op af en primitiv fortolker, en compiler, en editor, en assembler og en række primitive procedurer, som udgør det basale ordforråd.

Fortolkeren sørger for at dirigere processoren rundt i programmet. Den er ganske lille, idet den ikke har nogen traditionel fortolkerfunktion på den måde, at den ikke skal oversætte kommando-ord eller P-kode, men blot holde styr på en række absolutte adresser. Ej heller har den af samme grund nogen parser.

Da fortolkningsprocessen er meget enkel, vil et program meget hurtigt blive eksekveret, i modsætning til f.eks. BASIC-programmer, som gennemgår en temmelig kompliceret fortolkningsproces. I sammenligning med PASCAL vil FORTH ofte være hurtigere, idet PASCAL jo først skal oversætte sin P-kode, før en procedure kan identificeres og eksekveres, hvorimod FORTH direkte får adressen på sin procedure leveret (6).

Til dette formål benyttes compileren, der indeholder en parser, som kan operere på én linie ad gangen (input bufferen) eller på hele indholdet af tekst-editoren.

Når et ord defineres, afsættes der først plads til navnet (4 bytes), dernæst afsættes der plads til 2 vektorer (yderligere 4 bytes); den ene peger tilbage til det sidst definerede ord, den anden peger på koden for de rutiner, der medgår i definitionen af ordet.

Dette er det såkaldte definitionshovede, som altså består af 8 bytes, hvortil så kommer de (den) underordnede rutine(r):

Hvis et ord er defineret i FORTH vil hovedet være efterfulgt af adresserne på de i definitionen anvendte rutiner og nr. to vektor vil da pege på en rutine inde i fortolkeren, som vil dirigere processoren til de efterfølgende adresser.

Den sidste adresse peger på en anden rutine inde i fortolkeren, som sørger for at gå tilbage til compile mode eller videre til næste ord.

Hvis et ord derimod er defineret i maskinkode, vil denne oftest (men ikke altid) være anbragt lige efter definitionshovedet, hvorfor nr. to vektor i denne situation vil pege direkte på maskinkoden.

Fra compilerens synspunkt er definitionerne altså kædet sammen på en sådan måde, at den sidst skrevne peger tilbage på den næstsidste osv. Indtil den møder en vektor, som har værdien 0000, hvilket indikerer slut på kæden (se fig. 1).

Når compileren modtager et ord som del af definitionen på et nyt ord, og det er omgivet af et eller flere mellemrum (ASCII space) vil den søge igennem ordforrådet, indtil den finder ordet, og vil da hægte vektor nr. to på sin definition til senere brug for fortolkeren.

Hvis den ikke kan finde ordet, vil den prøve at lave en konstant ud af ordet i overensstemmelse med den gældende talnotation, og skubbe værdien på stack'en; såfremt dette ikke er muligt, vil der blive skrevet en fejlmeddelelse ud.

Som nævnt kan man også skrive sine definitioner i assembler. Da der oftest vil være tale om korte rutiner til f.eks. I/O eller optimering af den indre del af en løkke, er der til formålet kun skrevet en ganske simpel assembler (i øvrigt i FORTH), hvilket dog ikke hindrer linking af et større maskinkodeprogram, som i givet fald kan være skrevet i en større assembler.

Den indbyggede assembler er af enkelt-pass typen, hvor op-koderne er en del af FORTH's ordforråd. Dette har til følge at fremadrettede referencer ikke kan skrives ved hjælp af labels, hvilket, sammenholdt med at operanderne tilføres i omvendt polsk notation, gør assembler syntax'en besværlig at læse bagefter.

Men dette er uden praktisk betydning, idet der sjældent vil være ret mange assembleringsinstruktioner i et FORTH ord, og endelig er memory besparelsen værd at tage med: assembleren i 6502-versionen af FORTH (3) fylder kun 500-600 bytes alt inklusive!

Teksteditoren er til rådighed, hvis større definitioner end hvad en enkelt linie kan indeholde, er nødvendige i en given applikation.

FORTH eller assembler source code samt en blanding af disse kan her skrives ind og redigeres, inden de bliver kompileret. Det er også muligt via teksteditoren at gemme sin source kode i et baggrundslager, således at en fremtidig ændring er mulig.

Editorbufferen består af et felt på  $16 \times 64 = 1024$  karakterer, som er et passende format til de fleste skærmterminaler og tillige er kompatibel med de hyppigst anvendte masseopbevaringsmedier, såsom floppy disc, kassettebånd, magnetiske boble-hukommelser m.v.

Denne kombination kan i praksis fungere som et virtuelt lager, dog med kompilering for hver block, der bliver læst ind i teksteditoren. Kompileringen er til alt held meget hurtig, så det er ikke nogen større gene.

## Funktion

Nye ord, som bliver defineret, vil fra da af være en del af FORTH's ordforråd. Det er på den måde muligt at udvide sproget og skrive system- og applikationsspecifikke rutiner ind, som herefter kan anvendes i fremtidige software udviklinger.

Det er muligt at gruppere en del af ordforrådet i separate "dialekter", som før kompilering kan aktiveres og bagefter frakobles efter behov.

Assembleren og editoren er netop skrevet på denne måde, hvorved man opnår at kunne tillade en vis redundans med hensyn til navne, som ellers ville være i konflikt med hinanden. Et eksempel herpå er assembler op-koden "AND", der ellers kunne forveksles med det 16-bits logiske "AND", som findes i hovedordforrådet.

Når assembleren er aktiveret vil AND blive betragtet som op-kode, idet den sidst aktiverede dialekt vil blive gennemløbet først, og det basale ordforråd, som aldrig kan deaktiveres, vil blive gennemløbet til sidst:

Hver dialekt har i øvrigt en slutvektor =  $\emptyset\emptyset\emptyset$  (se fig. 2).

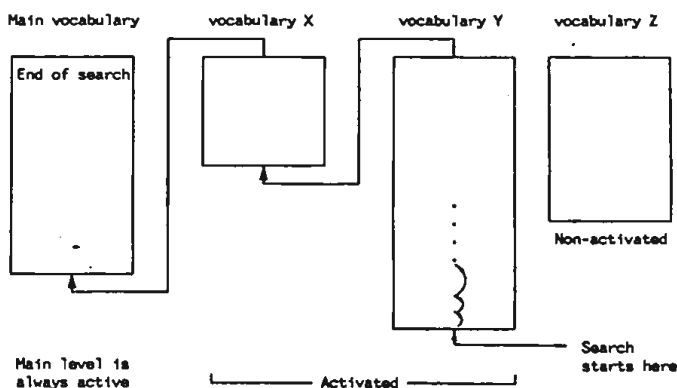


Fig. 2. Search sequence through active vocabulary

Det er vigtigt at slå fast, at aktivering/deaktivering af rutinerne i en dialekt kun har betydning ved kompilering, da rutinerne jo derefter kun refereres ved deres absolutte adresser, og derfor altid vil være anvendelige for det kompilerede program.

Der er altså tale om en slags "idiotsikring", som tillige har til formål at sætte programmøren i stand til at reducere kompileringstiden for større programpakker, ved kun at aktivere de nødvendige grupper ord.

Hvert ord i et program vil ofte være bygget op af flere underordnede ord, der igen kan være bygget op af andre underordnede ord. Derved opnås en hierarkisk eller pyramidestruktur (se fig. 3).

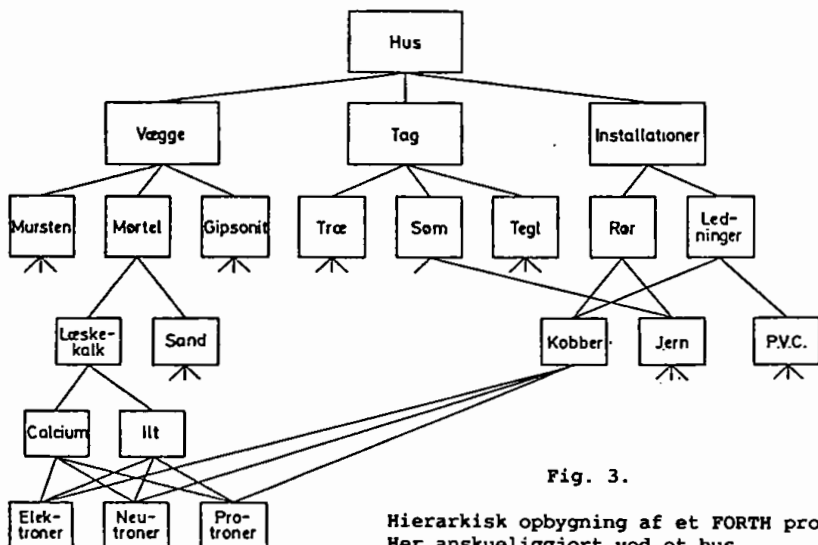


Fig. 3.

Hierarkisk opbygning af et FORTH program. Her anskueliggjort ved et hus. Bemærk, at jo længere nede i hierarkiet man bevæger sig, jo oftere vil en enhed være benyttet.

Da der heller ikke er nogen form for GOTO eller JUMP (der er jo ingen linienumre eller linietags) vil FORTH simpelthen tvinge programmøren til at designe "strukturerede programmer" (2). Den hierarkiske struktur gør det endvidere let at fejlfinde et program, idet underprogrammerne interaktivt kan testes ganske uafhængigt af helheden.

Endnu en fordel ved denne metode er, at definitionerne via teksteditoren kan gemmes i programmørens eget software-bibliotek til eventuelt senere brug i en anden opgave.

Til dannelse af betingelser og løkkestrukturer er det muligt inden for hvert ord at benytte PASCAL-lign. nøgleord som CASE, IF-ELSE-THEN, BEGIN-END, BEGIN-WHILE-REPEAT og FORTRAN-kendingen DO-LOOP, foruden en rekursiv operator til kald af egen definition.

I grundversionen af FORTH er kun følgende variabel-typer defineret: Konstanter, variable, arrays og matricer. Det er dog på simpel vis muligt at definere nye typer variable, idet systemet stiller både en funktion til rådighed til at definere variable i assembler og en til at definere i FORTH.

Det er derfor let at skrive sin egen validitetscheck til en ny datatype. Specielt til microprocessor-applikationer er det også rart at vide, at FORTH vil operere med tabaser fra 2 (binær) til 37 for hvem, der måtte have behov derfor!

Som allerede nævnt overføres parametre via en stack. Denne stack, som kaldes parameter stack'en, er en LIFO-type, men må ikke forveksles med microprocessorens returadresse stack, som stadig kan benyttes til subrutinekald i maskinkode. Der er imidlertid flere stacks i FORTH, af hvilke nogle er systembestemte og som regel usynlige for brugeren. Den vel nok mest anvendte af disse er retur stack'en, som bruges af systemet til løkke-strukturer og lignende.

Programøren har adgang til indenfor en definition at benytte denne stack til mellemresultater eller lignende, men alt anbragt på denne stack skal fjernes igen før proceduren forlades.

Endelig er dialekternes startadresser, d.v.s. det sidste ord i dialekten, gemt på en speciel stack, når de er aktiverede. Af denne grund kan man kun deaktivere dialekter i omvendt rækkefølge af aktiveringen.

### Anvendelsesområde

FORTH er blevet kritiseret for sin ulæselighed (4), hvilket dog kun gælder de lavere programmeringsniveauer, da højere ord i hierarkiet naturligt vil blive skrevet så de nøje beskriver den udførte funktion. Men "ulæseligheden" skyldes, at meget brugte ord bevidst er holdt nede på et lavt antal karakterer.

Endelig kan man også stille spørgsmålet, om et programmeringssprog skal være skrevet for den uøvede eller den erfarne programmør - FORTH vil være letlæseligt også på det mere primitive niveau for den erfarne programmør, men vil fremstå som volapyk for den, som kun har lært sig BASIC.

FORTH er til dato det eneste mini- og microcomputersprog som helt og aldeles understøtter "struktureret programmering" og "Top-down-design" (5), som prædikes af de fleste softwareguru'er. Af denne grund er sproget endog særdeles velegnet til undervisningsbrug, idet en programmør, som har en solid baggrund i FORTH naturligt vil skrive velstrukturerede programmer i et senere anvendt programmeringssprog.

De til dato udsendte microprocessor-versioner af FORTH benytter sig af heltalsaritmetik, hvorfor sproget p.t. først og fremmest vil finde anvendelse indenfor processtyring og dataopsamlingsområdet.

Det vil dog ikke være nogen uoverkommelig opgave at hægte en flydende-komma matematikpakke på, ved simpelthen at redefinere de matematiske operatører. En alternativ mulighed består i at benytte sig af dobbelte eller tredobbelte ord til specielle formål. Dette vil være meget lidt tidkrævende.

Specielt til større programmeringsopgaver vil det være gavnligt at anvende FORTH. Jo større et problems kompleksitet, jo større fordel vil der være ved at anvende FORTH med hensyn til forbrug af memory (se fig. 4).

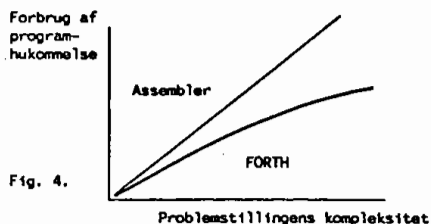


Fig. 4.

Det vil derfor ofte i en microprocessor-opstilling være spørgsmålet om at behøve en disc eller at kunne klare sig med de væsentlig billigere og mere pålidelige (E)Prom'er.

Noget nær det eneste FORTH ikke kan bruges til er at udrydde den konservatisme med hensyn til valg af programmeringssprog, som også programmører er i besiddelse af.

Jeg håber, det ikke lykkes i de "bedrevidendes" hellige navn, at sable en god ide ned, en ide som den danske elektronikindustri kan have god brug for i disse tider. FORTH behøver ikke statsstøtte som sit operativsystem!

E.S. Richmond, BC, Canada.

#### Litteraturliste

1. E.D. Rather  
& C.H. Moore: *The FORTH Approach to operating Systems.*
2. J.D. Warnier: *Logical Construction of Programs,*  
H.E.Stenfert Krose, Leiden (1974).
3. R. Deane og E.C. Rehnke: *6502 FORTH User Manual,*  
Eric C. Rehnke, CA 92807 (1979).
4. R.D.Grappel: *STRUBAL VS. FORTH,* Dr. Dobb's Journal, Vol 3, issue 8 1978)
5. E.D.Rather  
& C.H.Moore: *FORTH High-level programming Techniques on Microprocessors.*
6. J. R. Bell: *Threaded Code,* Comm. ACM, Vol. 16, No. 6 (1973)
7. M. Mannoni: *FORTH - An Extensible Path to Efficient Programs.*  
Electronic Design, Vol. 28, No. 15, side 175-178, (1980).
8. A. Taylor: *FORTH Becoming Hothouse for Developing Languages.*  
Computerworld, 30 juli 1979.

#### PC-biblioteket:

Biblioteket råder pt. over PC-SIG numrene 1-666, dog med undtagelse af nedenstående. Foruden de 'alm. volumes' kan også leveres et sæt index disketter (2), de indeholder en beskrivelse som den, der vises i bladet. Pris for index disketterne er 50 kr.

Manko:

|     |     |         |     |     |     |
|-----|-----|---------|-----|-----|-----|
| 032 | 145 | 346-350 | 400 | 516 | 602 |
| 060 | 149 | 352     | 488 | 518 | 603 |
| 063 | 150 | 354-359 |     | 529 |     |
|     | 151 | 361     |     | 541 |     |
|     |     | 363-373 |     | 561 |     |
|     |     | 376     |     |     |     |
|     |     | 378     |     |     |     |

. . . kan nogen hjælpe biblioteket med ovenstående PC-SIG numre?

#### ANNONCE

Sælges: C.Itoh type 8300F printere ( 2 stk.) á 250 kr.  
5 x 7 matrix. 80 cols., seriel inp. 110-4800 bps

SHUGART 8" diskdrev SS SD ( 2 stk.) á 200 kr.  
Alle dele er brugte, men ok.

Chr. Hemdrup, 86 22 89 38, Gl. Kirkevej 39 B, 8530 Hjortshøj

## SOFTWARE & DISKETTER

CP/M-volumes bestilles ved CP/M-bibliotekaren.

Husk ved bestilling af CP/M-volumes at oplyse om diskformat.

PC-volumes bestilles ved PC-bibliotekaren.

Volume fra bibliotek (5.25") incl. disk & forsendelse 30,- kr.

OBS! *Nye telf. (område) numre!*

----- Bestyrelsen: -----

### Formand:

-----  
Donald Axel  
Saxenkolvej 20  
3210 Vejby  
42 30 69 13

-----  
Niels Veileborg  
Finsensvej 11B  
2000 Kbh. F.  
31 87 80 85

-----  
Henrik Persson  
Niels Lorentzensvej 58  
Blistrup  
3230 Græsted  
42 11 57 76

### Kasserer:

-----  
Lars Gråbæk  
Esbern Snaresgade 6  
1725 Kbh. V.  
31 23 92 36

-----  
Poul Bigum  
Ved Hegnet 1  
2100 Kbh. Ø.

-----  
Poul Waltoft  
Kronborggade 26  
2200 Kbh. N.  
31 85 34 51

-----  
Carl Skovgaard Larsen  
Pærevangen 19  
2765 Smørum  
44 66 11 27

### CP/M Bibliotek:

-----  
Jørgen Petersen  
Sofiegade 24, 2  
1418 Kbh. K.  
31 54 91 76  
(man-tor 20-21)  
Giro 1 92 80 66

### PC-Bibliotek:

-----  
Peter Rasmussen  
Strandengen 9  
4000 Roskilde.  
( skriftligt )  
Giro 7 49 91 40

### Bulletin Board:

-----  
Telf. 42 11 77 29  
Åbent hele døgnet  
300, 1200/75, 1200,  
2400 bits/sec, 8 bit,  
1 stop, ingen paritet

### Disketteredaktør:

Carl Skovgaard Larsen

-----  
Redaktør: Viggo Jørgensen, FensmarksAlle 6, 3520 Farum, 42 95 32 01

